Infrastructure

behind distributed systems

micro-23

Aliaksei Bialiauski

Designed in LaTeX

Infrastructure components

Design in the Cloud

PaaS, IaaS, SaaS, EaaS, etc.

Chapter #1:

Infrastructure components

Infrastructure

Server components

- CPU(easy to scale)
- Memory
- Disk
- Network(hard to scale)

Rack

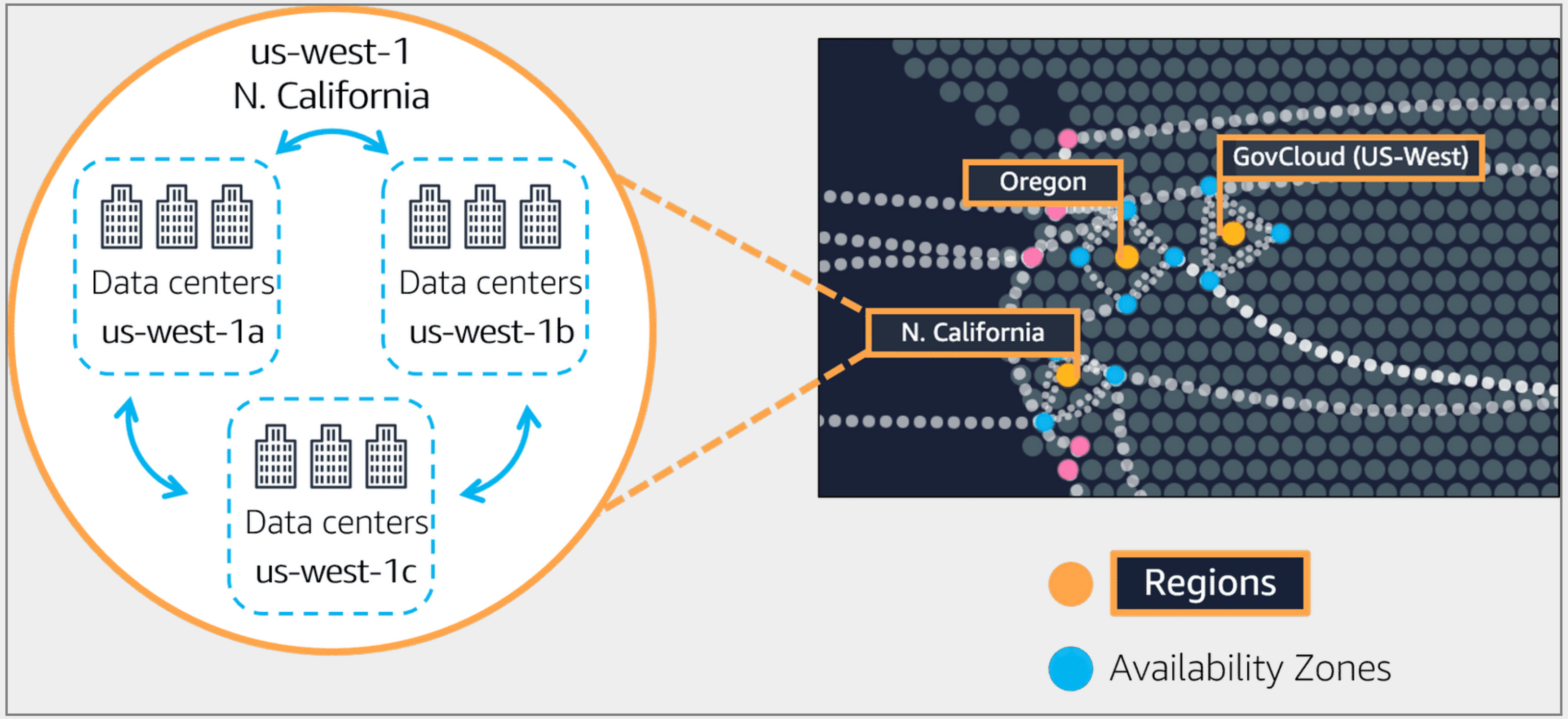
Rack - a set of servers.

Data Centers

DC - a set of server racks.

May become unavailable due to power outage, earthquake, and other disaster.

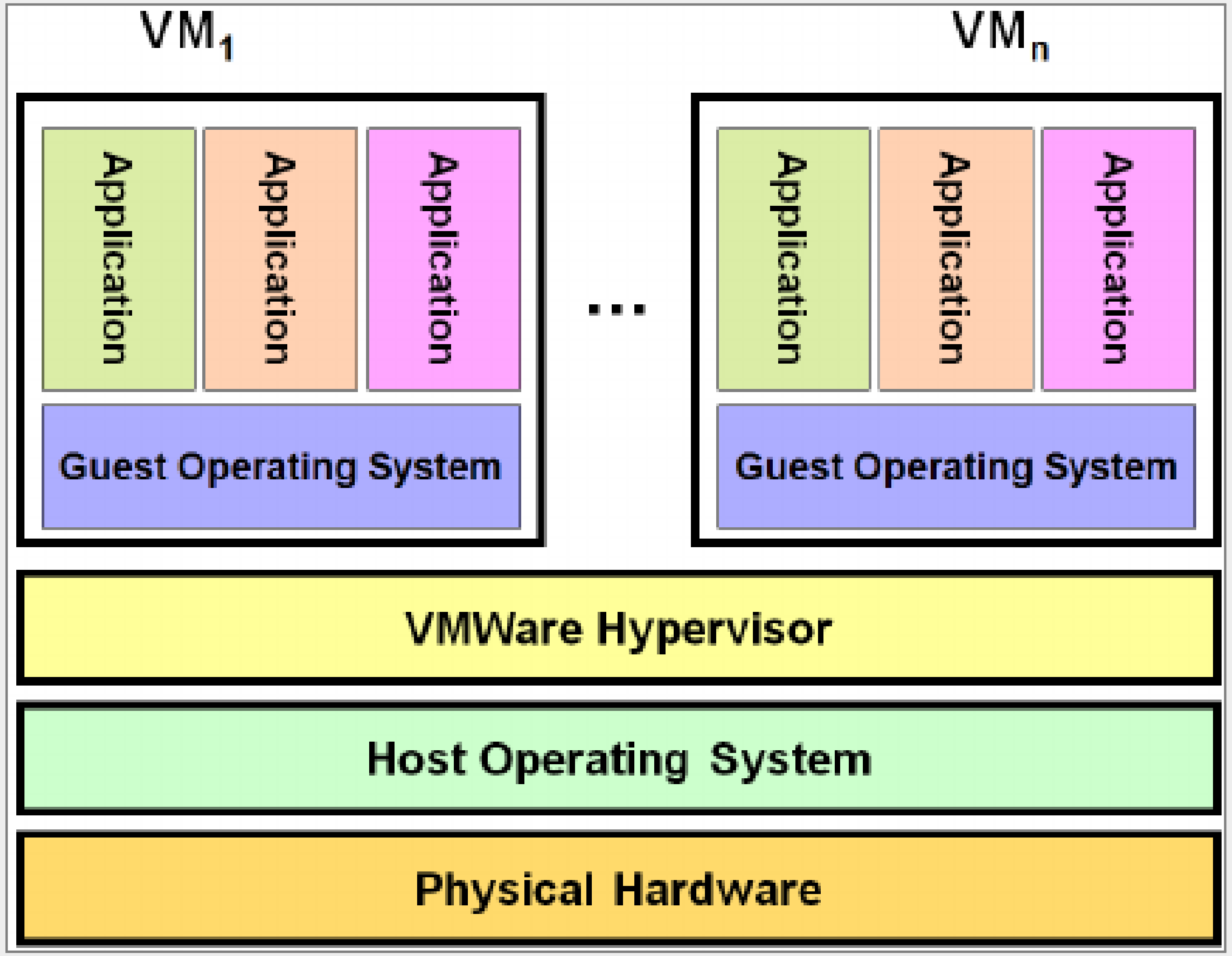
Availability Zone



Regions

Region - a network of 2 AZs.

https://awsregion.info


VM_1
VM_n
Application
Application
Application
Guest Operating System
...
Application
Application
Application
Guest Operating System
VMWare Hypervisor
Host Operating System
Physical Hardware

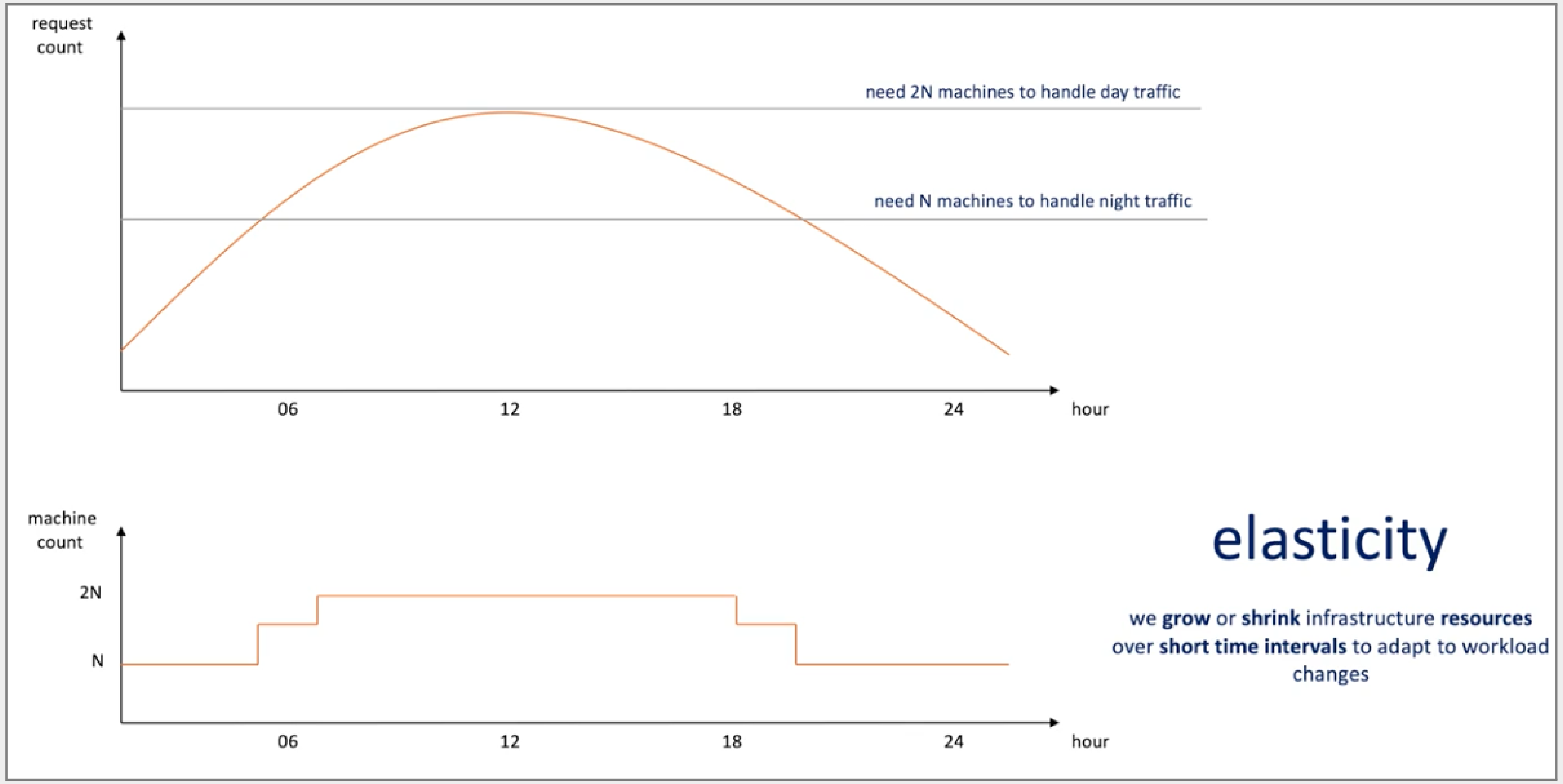
Containers

"A container is a lightweight, standalone, executable package of software that includes everything is needed to run an app: code, settings, runtime, system tools and system libraries." — GCP docs.

Chapter #2:

Design in the Cloud

Autoscaling - corner stone of Cloud Computing



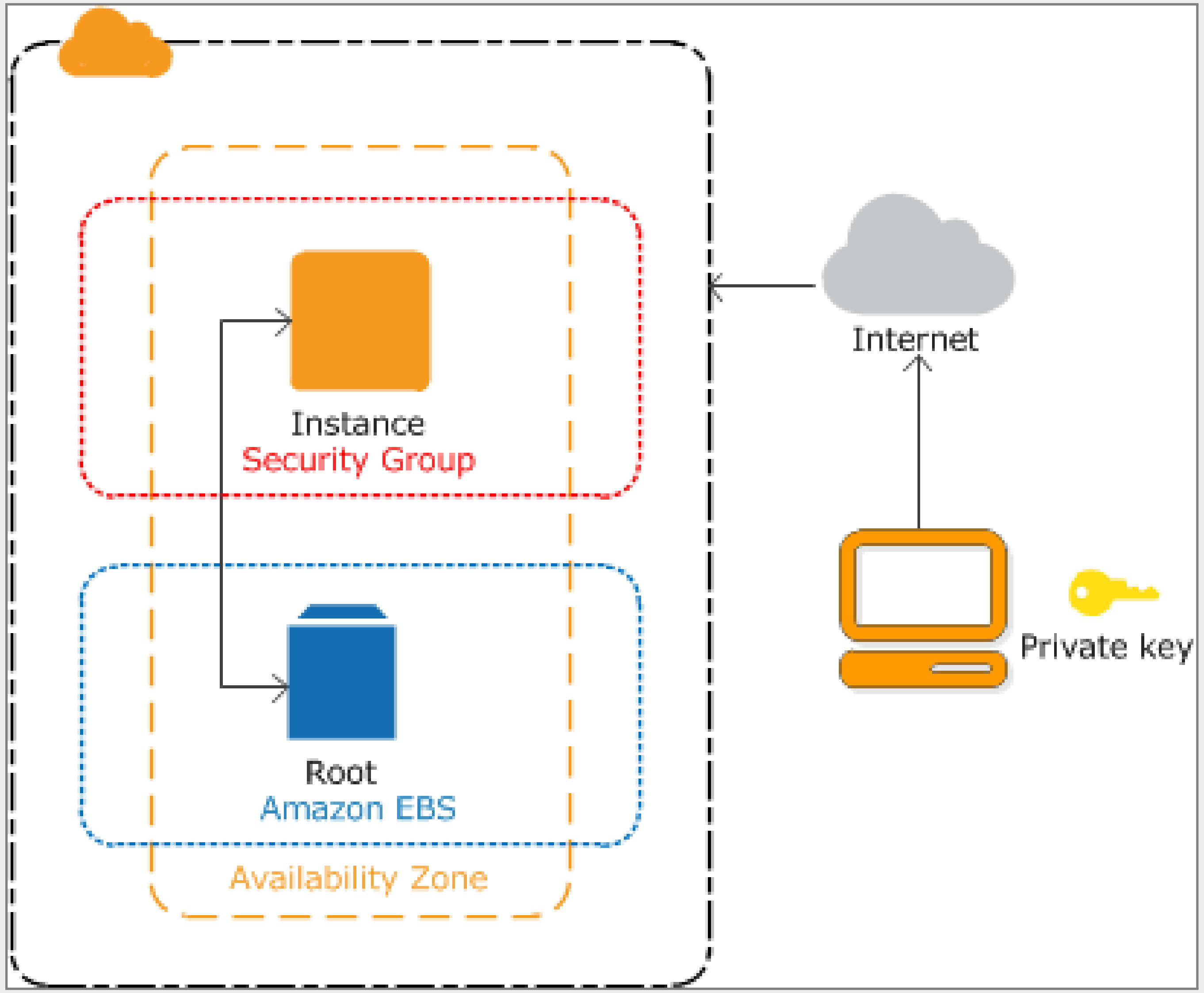
elasticity

we **grow** or **shrink** infrastructure **resources** over **short time intervals** to adapt to workload changes

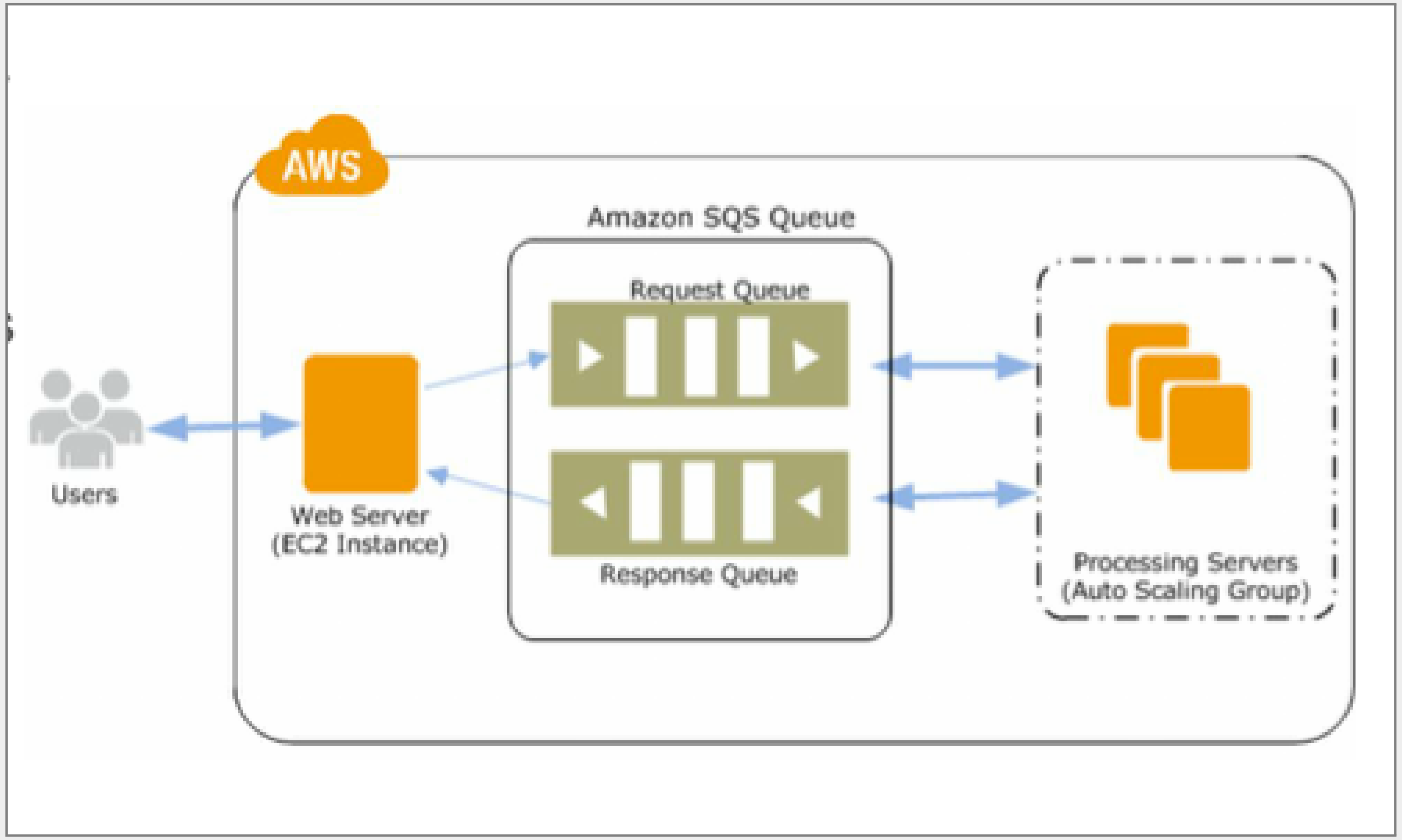
S3

```
class S3 {
  public static void main(String[] args) {
    AWSCredentials credentials = new BasicAWSCredentials(
      "<AWS accesskey>",
      "<AWS secretkey>"
    );
    AmazonS3 s3 = AmazonS3ClientBuilder.standard()
      .withCredentials(new AWSStaticCredentialsProvider(credentials))
      .withRegion(Regions.US_EAST_2)
      .build();
    s3.createBucket("user_images");
    s3.putObject(
      bucketName,
      "Document/jeff.png",
      new File("jeff.png")
    );
  }
}
```

EC2

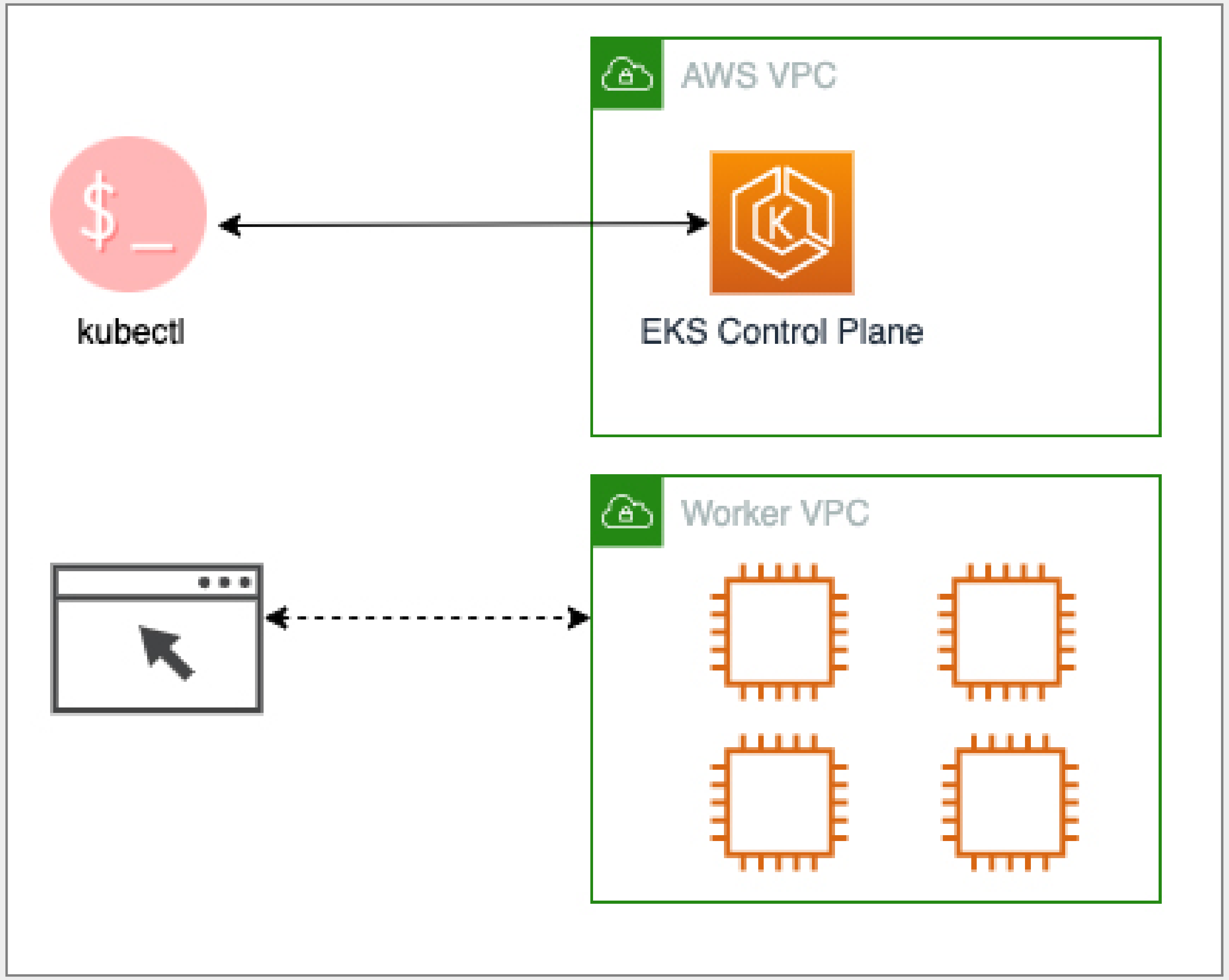

Instance
Security Group
Root
Amazon EBS
Availability Zone
Internet
Private key

SQS

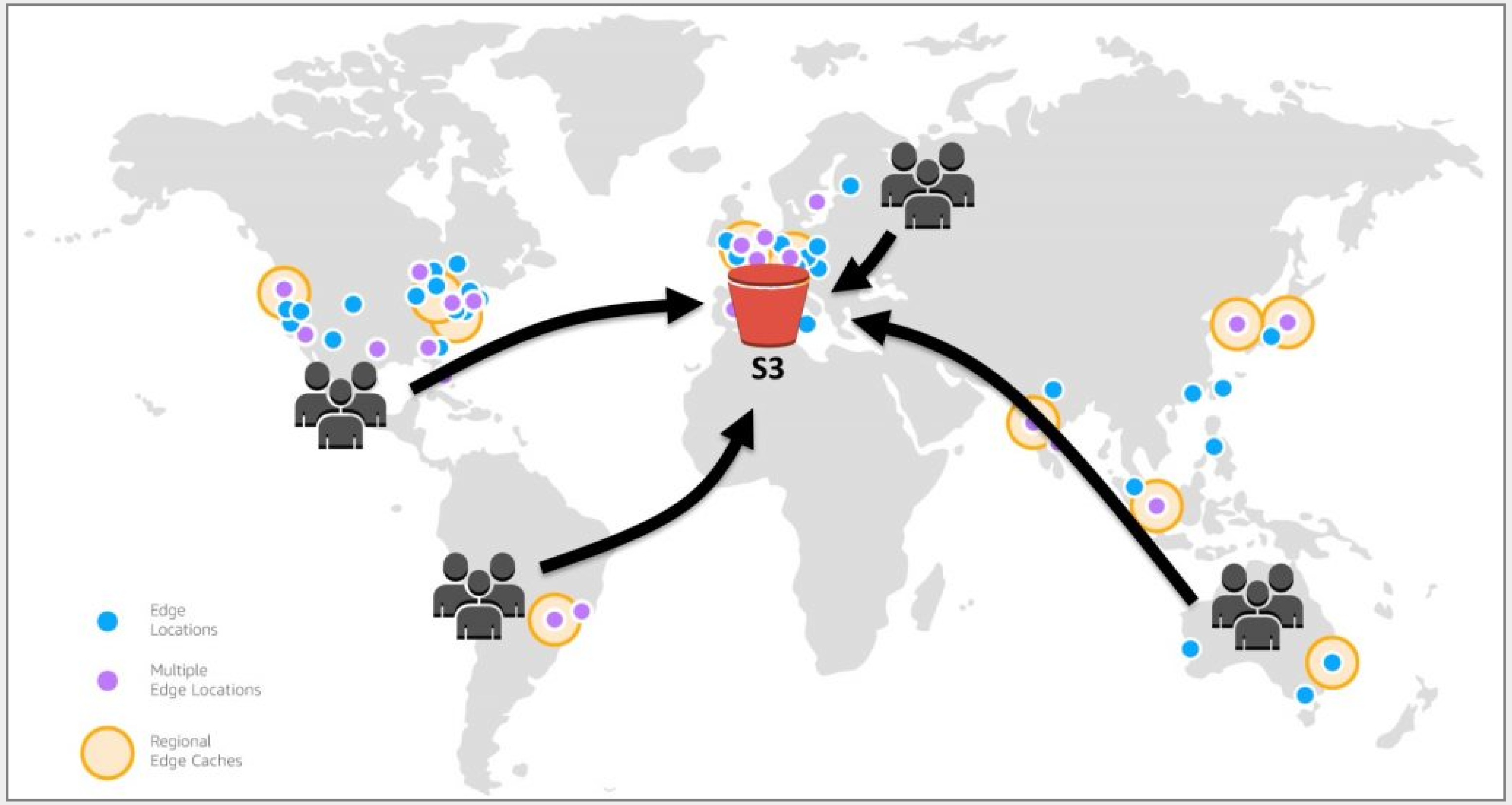

AWS
Amazon SQS Queue
Request Queue
Response Queue
Users
Web Server
(EC2 Instance)
Processing Servers
(Auto Scaling Group)

RDS

EKS - Managed Kubernetes Master



CloudFront


S3
Edge Locations
Multiple Edge Locations
Regional Edge Caches

Serverless architecture

Serverless architectures are application designs that incorporate third-party "Backend as a Service" (BaaS) services, and/or that include custom code run in managed, ephemeral containers on a "Functions as a Service" (FaaS) platform.

Fundamentally, FaaS is about running backend code without managing your own server systems or your own long-lived server applications.

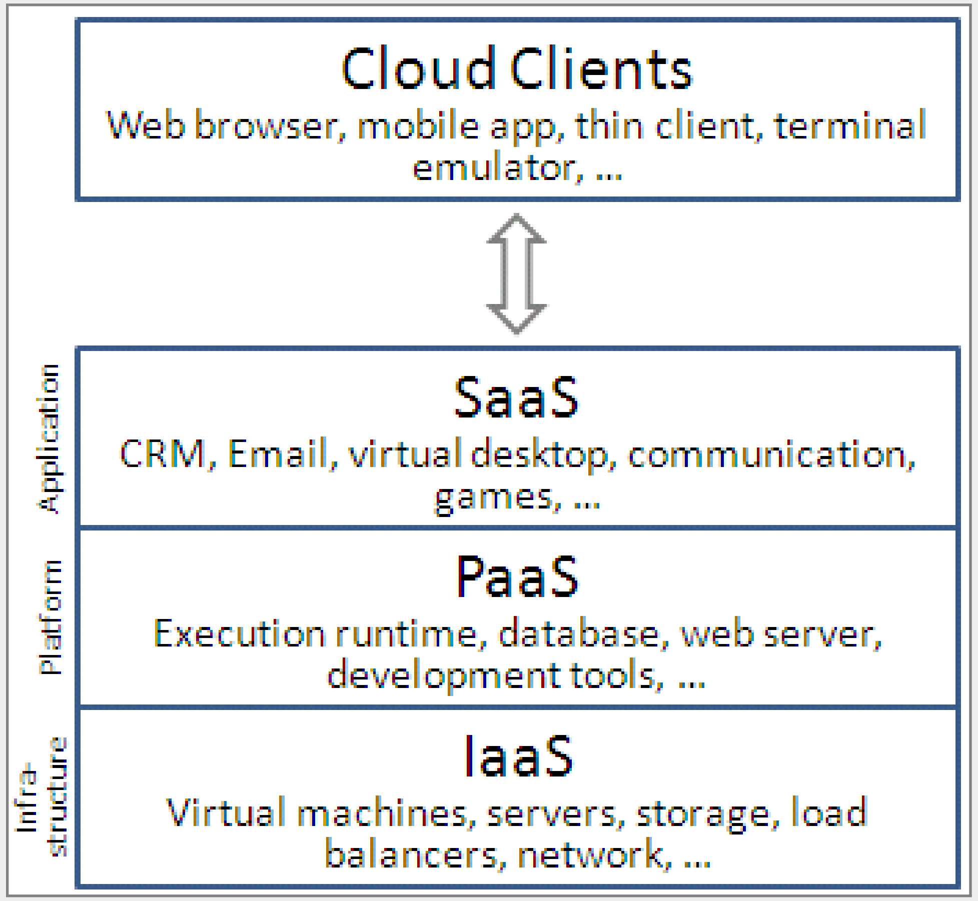
AWS RDS vs. AWS DynamoDB

https://martinfowler.com/articles/serverless.html

Chapter #3:

PaaS, IaaS, SaaS, EaaS, etc.

PaaS, IaaS, SaaS, EaaS, etc.



Cloud Clients

Web browser, mobile app, thin client, terminal emulator, ...

SaaS

CRM, Email, virtual desktop, communication, games, ...

PaaS

Execution runtime, database, web server, development tools, ...

IaaS

Virtual machines, servers, storage, load balancers, network, ...